



Energy Constrained Depth First Search

Shantanu Das¹ · Dariusz Dereniowski² · Przemysław Uznański³

Received: 4 June 2019 / Accepted: 20 September 2024 / Published online: 12 October 2024
© The Author(s) 2024

Abstract

Depth first search is a natural algorithmic technique for constructing a closed route that visits all vertices of a graph. The length of such a route equals, in an edge-weighted tree, twice the total weight of all edges of the tree and this is asymptotically optimal over all exploration strategies. This paper considers a variant of such search strategies where the length of each route is bounded by a positive integer B (e.g. due to limited energy resources of the searcher). The objective is to cover all the edges of a tree T using the minimum number of routes, each starting and ending at the root and each being of length at most B . To this end, we analyze the following natural greedy tree traversal process that is based on decomposing a depth first search traversal into a sequence of limited length routes. Given any arbitrary depth first search traversal R of the tree T , we cover R with routes R_1, \dots, R_l , each of length at most B such that: R_i starts at the root, reaches directly the farthest point of R visited by R_{i-1} , then R_i continues along the path R as far as possible, and finally R_i returns to the root. We call the above algorithm *piecemeal-DFS* and we prove that it achieves the asymptotically minimal number of routes l , regardless of the choice of R . Our analysis also shows that the total length of the traversal (and thus the traversal time) of *piecemeal-DFS* is asymptotically minimum over all energy-constrained exploration strategies. The fact that R can be chosen arbitrarily means that the exploration strategy can be constructed in an online fashion when the input tree T is not known in advance. Each route R_i

The work has been announced at the 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018) [13]

S. Das: Author partially supported by ANR project ANCOR (anr-14-CE36-0002-01)

D. Dereniowski: Partially supported by National Science Centre (Poland) grant number 2015/17/B/ST6/01887.

✉ Dariusz Dereniowski
deren@eti.pg.edu.pl

Shantanu Das
shantanu.das@lis-lab.fr

Przemysław Uznański
puznanski@cs.uni.wroc.pl

¹ LIS Aix-Marseille University University of Toulon CNRS, Marseille, France

² Gdańsk University of Technology, Gdańsk, Poland

³ Institute of Computer Science, University of Wrocław, Wrocław, Poland

can be constructed without any knowledge of the yet unvisited part of T . Surprisingly, our results show that depth first search is efficient for energy constrained exploration of trees, even though it is known that the same does not hold for energy constrained exploration of arbitrary graphs.

Keywords DFS traversal · Distributed algorithm · Graph exploration · Piecemeal exploration · Online exploration

1 Introduction

Graph-theoretic problems in which one wants to cover the entire graph with one or more routes satisfying certain objective is a well established and long studied topic in many areas of computer science. Particular problems vary depending on the research area or potential applications, including the study of simple graph traversals like DFS or BFS, for algorithmic purposes, to complex transportation problems with many variations of traveling salesman problem (TSP), or pursuit-evasion games like the watchman problem, and finally distributed monitoring of networks using mobile agents.

For one possible application of our results, consider a mobile robot that needs to explore an initially unknown tree. We assume that the tree is edge-weighted and the weight of each edge denotes the length of that edge. Starting from a single vertex (the root) of the tree, the robot must traverse all edges and return to its initial location. Upon visiting a vertex v for the first time, the robot discovers the edges incident to v and can choose one of them to continue the exploration. Provided that the robot can remember the visited vertices and edges, a simple depth first search (DFS) is an efficient algorithm for exploring the tree, achieving the optimal cost of twice the sum of the lengths of edges in the tree.

Consider a more interesting scenario, when the robot has a limited source of energy (e.g. a battery) which allows it to traverse a path of length at most B (we say such a robot is *energy constrained*). Naturally, we assume that each vertex of the tree is at distance at most $B/2$ from the root, otherwise the tree cannot be fully explored. In this case, exploration is possible if the robot can recharge its battery whenever it returns back to the starting location. Thus, the exploration is a collection of routes of the robot, each of which starts and ends at the root, and has length at most B . We are interested in the minimum number of such routes needed (i.e. the number of times the robot has to recharge) to completely explore the tree.

This model of exploration may be of interest for several reasons. One obvious reason is related to the capabilities of the robot; it may have a restricted fuel tank capacity or perhaps a harsh or risky environment enforces a return to its home-base every so often. A robot that returns periodically to the root can inform about new discoveries—in this way the knowledge is accumulated gradually at the base-station while the algorithm progresses. This may, for example, reduce the risk of having no data in case of robot failure prior to the end of exploration. From a different point of view, this process may be seen as a *piecemeal learning*, that is, one in which it is possible to have a trade off between exploration and utilization; the two phases representing parts in which learning occurs (exploration) and parts in which accumulated knowledge is

used (utilization). Finally, having many restricted-length routes covering a tree instead of a single long route may be potentially applied in scenarios in which one wants to minimize exploration time by strategies using multiple robots. In fact, when the robots are incapable of refueling, we can use several robots to explore the tree, each robot traversing a path of length B . In that case, it is important to minimize the number of robots used as well as the total energy cost for exploration.

Note that the *piecemeal exploration* problem has been studied before not just for trees but also for arbitrary connected graphs. However those results were restricted to visiting vertices at depth of at most $B/(2(1 + \beta))$, for some $\beta > 0$, with the cost of exploration deteriorating sharply as β approached zero, see e.g. [8]. In this paper we would like to consider exploration strategies that completely visit all trees up to the maximum possible depth of $B/2$. No such exploration algorithms have been studied for either general graphs or special graphs such as trees. Simple strategies based on depth-first search (breadth first search) perform badly in the case of piecemeal exploration of arbitrary graphs. However as we show in this paper, the piecemeal version of depth-first search performs optimally in trees. This fact is surprising given the fact that for exploration by multiple open routes (routes that do not end at the root) depth-first strategies in trees can have an overhead of $\Omega(\log n)$ [12].

Related work: There exists extensive literature on graph traversal and exploration, we survey here only the most relevant results on graph exploration by mobile agents (for a more extensive survey please see [11] and the references therein). Exploration of general graphs having n nodes and m edges, by a single agent, has been studied in [27] who gave an algorithm of $m + O(n)$ steps. For exploration by k agents, [22] provides an exploration algorithm taking $O(D + n/\log k)$ steps in trees of n nodes and height D . This algorithm turns out to be $O(k/\log k)$ competitive [24] (where competitiveness is the ratio of the number of steps of an algorithm over the optimal number of steps). Authors in [9] give a $O(n/k + D^{k-1})$ time algorithm for tree exploration while [20] gives an algorithm for sparse trees with competitive ratio $O(D^{1-1/p})$, where p is defined as the tree density. For some lower bounds on exploration time see [21, 22, 24]. For other recent results on exploration time see e.g. [9, 14, 16, 17, 25, 26]. Other than optimizing time, exploration using little memory for the agents has also been studied, see e.g. [1, 15].

None of the results mentioned above consider any energy limitation for the agents. The energy constrained exploration problem was first studied under the name of *Piecemeal Graph Exploration* [8], with the assumption that the route length $B \geq 2(1 + \beta)r$, where r is the furthest distance from the starting node to other nodes, and $0 < \beta < 1$. That paper provided exploration algorithms for a special class of grid graphs with ‘rectangular obstacles’. Awerbuch et al. [4] showed that, for general graphs, there exists an energy constrained exploration algorithm with a total cost of $O(m + n^{1+o(1)})$. This has been further improved (by an algorithm that is a combination of DFS and BFS) to $O(m + n \log^2 n)$ in [5]. Finally [18] provided an exploration algorithm for general unknown weighted graphs with total cost asymptotic to the sum of edge weights of the graph. Note that, as mentioned, all the above strategies require the length of each route to be strictly larger than the shortest return path from the starting vertex to the farthest vertex. In other words, these algorithms fail in the extreme cases when the height of the

explored tree (or the diameter of the graph) is equal to half of the energy budget, which seem to be the most challenging cases. When the tree is known in advance, finding the offline optimal strategy for piecemeal exploration is known to be NP-complete for weighted trees [22].

The same tree exploration model as we study in this work has been considered in [6, 12, 19] for unweighted trees and multiple agents, with one difference: each agent traverses a path of length at most B such that the path starts at the root but may end at any node of the tree (in other words, agents do not have to return to the homebase). It has been shown in [12] that if the tree is not known in advance, then there exists an exploration algorithm (that minimizes the number of agents used) with competitive ratio of $O(\log B)$ and this is the best possible. On the other hand, it was shown that by allowing the route lengths to be a constant factor more than B , it is possible to explore the tree using the minimum number of agents [19]. Distributed algorithms for energy constrained agents have been a subject of recent investigation, see e.g. [2, 7]. Authors in [10] also consider a model in which agents are allowed to transfer part of their energy to another agent. There are many studies on variants of the traveling salesman problem, including the k -TSP [3, 23] related to the task of finding a bounded length route in a graph. Such results are out of scope for this paper.

Our Results and Outline: In this work we analyze a very natural process of partitioning a depth first search traversal R_{DFS} of a tree into a sequence $\mathcal{S} = (R_1, \dots, R_k)$ of routes where each route R_i has length at most B , starts and ends at the root of the tree (see Sect. 3 for a formal definition). We prove that the number of routes k is asymptotically optimal (Theorem 1), that is, it is within a constant factor of the number of routes in any exploration strategy composed of routes of length at most B that cover the entire tree. This fact, being intuitively expected for trees (although it does not hold in general graphs [18]) turns out to be nontrivial. Our approach is to consider another parameter of an exploration strategy: the *cost* (see Sect. 4) defined as the sum of the lengths of all routes in an exploration strategy. In particular, we show in Sect. 4 that the fact that \mathcal{S} has small cost implies that the number of routes in \mathcal{S} is expectedly small. In order to prove our main result, we argue, in Sect. 5, that the cost of \mathcal{S} is asymptotically optimal (see Theorem 2).

We emphasize that the above claim holds independently of the choice of the initial depth first search traversal R_{DFS} . The implications of this fact are twofold. First, it provides a theoretical insight into such a partitioning of a depth first search traversal into bounded-length segments. Second, for an exploration algorithm design it means that the routes R_i may be constructed without knowing R_{DFS} in advance, or more precisely, the routes may be built in an online fashion based only on the knowledge of the subtree explored to date. This property makes our algorithm suitable for online exploration of unknown trees by energy constrained mobile agents.

This result also makes it straightforward to make an easy linear-time implementation of an algorithm for finding such bounded-length tree covering routes R_i : it is enough to find any DFS traversal of the input tree and then obtain the R_i 's by following the formal definition in Sect. 3.

2 Exploration Strategies

In this work we consider edge-weighted rooted trees $T = (V(T), E(T), \omega: E(T) \rightarrow \mathbb{R}_+)$, with root r . We define a *route* R as sequence $R = (v_0, v_1, \dots, v_l)$, where v_i is a vertex of T for each $i \in \{0, \dots, l\}$, as follows:

- (i) $\{v_i, v_{i+1}\} \in E(T)$ for each $i \in \{0, \dots, l-1\}$,
- (ii) $v_0 = v_l$ is the root r of T .

Informally speaking, a route is a sequence of vertices forming a walk in T that starts and ends at the root. We define the *length* of R to be

$$\ell(R) = \sum_{i=1}^l \omega(\{v_{i-1}, v_i\}).$$

We say that a vertex v is *visited* (and an edge $\{v_{i-1}, v_i\}$ is *traversed*) by the route if $v = v_i$ for some $i \in \{1, \dots, l\}$. We also say that the subtree of T composed with all vertices visited by R is *covered* by the route.

Given a tree T and a real number B , we say that $\mathcal{S} = (R_1, \dots, R_k)$ is a *B-exploration strategy* for T (or simply *exploration strategy* if B is clear from the context) if for each $i \in \{1, \dots, k\}$, R_i is a route in T of length at most B , and each vertex of T is visited by some route in \mathcal{S} . We write $|\mathcal{S}|$ to refer to the number of routes in \mathcal{S} , $k = |\mathcal{S}|$.

The *height* of a tree is the maximum weighted length of all paths between the root and the leaves.

3 Problem Statement and DFS Exploration

The formulation of the combinatorial problem, to which we refer as *energy constrained tree exploration*, we study in this work is as follows.

Energy Constrained Tree Exploration problem (ECTE)

Given a real number $B > 1$ and an edge-weighted rooted tree T of height at most $B/2$ what is the minimum integer k such that there exists a B -exploration strategy that consists of k routes?

Our goal is to analyze a particular type of solution to this problem, namely, an exploration strategy that behaves like a depth first search traversal but adapted to the fact that route lengths are bounded by B . Let for the rest of the paper $R_{\text{DFS}} = (v_0, v_1, \dots, v_l)$ be a route in T that covers the tree T and performs a depth first search traversal of T . (Note that R_{DFS} is a route and thus we consider a depth first search traversal to have node repetitions.) For two vertices u and v of T , $d(u, v)$ denotes the distance between u and v understood as the sum of weights of the edges of the path connecting these vertices. We refer by $\text{PDFS}(T) = (R_1, \dots, R_k)$ (*Piecemeal Depth First Search*) to the following B -exploration strategy constructed iteratively for $i := 1, \dots, k$ (see also Fig. 1 for an example):

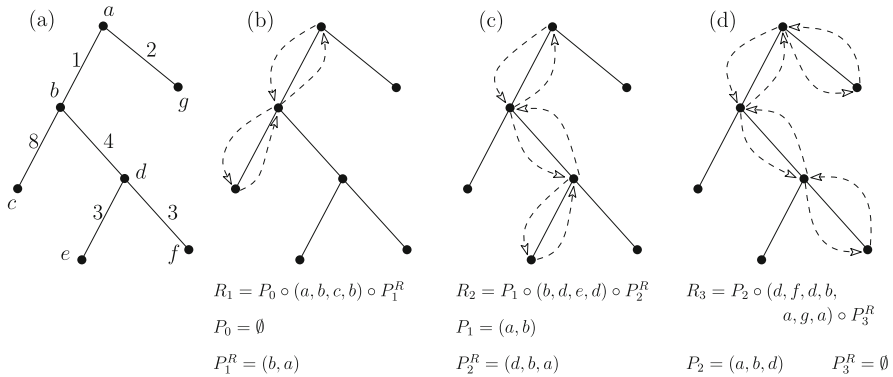


Fig. 1 A route R_{DFS} and the corresponding $\text{PDFS}(T) = (R_1, R_2, R_3)$ with $B = 20$: **a** a depth first search traversal $R_{\text{DFS}} = (a, b, c, b, d, e, d, f, d, b, a, g, a)$; **b–d** routes R_1, R_2, R_3 with lengths 18, 16 and 20, respectively

- (i) let $j_0 = 0$, i.e., $v_{j_0} = v_0 = r$,
- (ii) R_i continues the DFS exploration from where R_{i-1} stopped making progress (from the node $v_{j_{i-1}}$) as long as for the currently visited v_p :

$$d(r, v_{j_{i-1}}) + \ell((v_{j_{i-1}}, v_{j_{i-1}+1}, \dots, v_p)) + d(v_p, r) \leq B, \quad (1)$$

- (iii) the farthest v_p (for $p \leq l$) that satisfies the condition in (ii) is denoted as v_{j_i} , the vertex where R_i stopped making progress,
- (iv) let $R_i = P_{i-1} \circ (v_{j_{i-1}}, v_{j_{i-1}+1}, \dots, v_{j_i-1}, v_{j_i}) \circ P_i^R$, where P_{i-1} is the path from r to $v_{j_{i-1}}$, and P_i^R is the path from v_{j_i} to r .

Such a strategy $\text{PDFS}(T)$ is called a *DFS B-exploration*. We will say that the part of R_i containing the subsequence $(v_{j_{i-1}}, \dots, v_{j_i})$ makes progress on the route R_{DFS} .

We remark that different depth first search traversals R_{DFS} may result in different values of k (different number of routes) in the resulting DFS B -exploration, although for a particular choice of R_{DFS} the corresponding $\text{PDFS}(T)$ is unique. In the rest of the work we fix the route R_{DFS} arbitrarily and thus $\text{PDFS}(T)$ refers to the unique DFS B -exploration strategy obtained from this R_{DFS} .

We also remark that, although DFS-piecemeal exploration strategies we consider are asymptotically optimal for weighted trees, they are not necessarily optimal as shown in an example in Fig. 2.

4 Our Results

The following theorem provides the first main result of this work.

Theorem 1 *Let T be a tree and let the longest path from the root to a leaf in T be at most $B/2$. It holds $|\text{PDFS}(T)| \leq 12 |\mathcal{R}|$, where \mathcal{R} is a B -exploration strategy that consists of the minimum number of routes.*

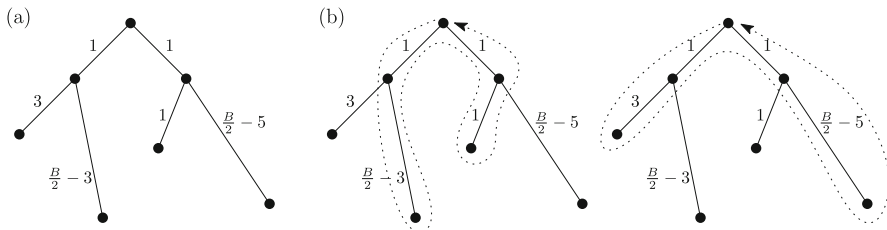


Fig. 2 DFS-piecmearl exploration needs not be optimal: **a** a weighted tree with $B > 10$; **b** two routes that provide an optimal B -exploration strategy and do not form a piecmearl DFS (each $\text{PDFS}(T)$ contains more than two routes)

The theorem refers to the number of routes in an exploration strategy. However, in order to analyze the behavior of $\text{PDFS}(T)$, we will work with another parameter on which we will focus in the entire analysis in the subsequent section. For any B -exploration strategy $\mathcal{S} = (R_1, \dots, R_k)$ of T we will denote by $\xi(\mathcal{S})$ the *cost* of \mathcal{S} defined as

$$\xi(\mathcal{S}) = \sum_{i=1}^k \ell(R_i).$$

We denote by $\text{COPT}(T)$ an optimal solution with respect to the cost, that is, a B -exploration strategy whose cost is minimum over all B -exploration strategies. This strategy will serve as a reference point to prove asymptotic optimality (in terms of the number of routes) of the DFS exploration. More precisely, we will prove the following theorem.

Theorem 2 *Let T be a tree and let $B/2$ be greater than or equal to the longest path from the root to a leaf in T . It holds $\xi(\text{PDFS}(T)) \leq 12 \cdot \xi(\text{COPT}(T))$.*

The proof is postponed to the next parts of the paper and we finish this section by concluding that Theorem 2 indeed implies Theorem 1.

Proof of Theorem 1 We start with the following observation which relates the smallest possible number of routes in a B -exploration strategy and the minimum possible cost.

observation 3 *Given T and B , $|\mathcal{R}| \geq \lceil \xi(\text{COPT}(T))/B \rceil$, where \mathcal{R} is a B -exploration strategy with minimum number of routes.*

Proof Each route of \mathcal{R} is of length at most B . Thus, $|\mathcal{R}| \geq \xi(\mathcal{R})/B$. By definition of $\text{COPT}(T)$, $\xi(\mathcal{R}) \geq \xi(\text{COPT}(T))$, and since $|\mathcal{R}|$ is an integer, the claim follows. \square

Recall that $R_{\text{DFS}} = (v_0, \dots, v_l)$ is the depth first search traversal of T used to obtain $\text{PDFS}(T)$, and the i -th route R_i in $\text{PDFS}(T) = (R_1, \dots, R_k)$ makes progress on the depth first search traversal by traversing the part of R_{DFS} that starts at $v_{j_{i-1}}$ and ends at v_{j_i} . By definition of $\text{PDFS}(T)$, extending R_i so that it makes progress with the walk $(v_{j_{i-1}}, \dots, v_{j_i}, v_{j_{i+1}})$ would exceed its length to be more than B for each $i < k$, i.e., $\ell(R_i) + 2\omega(\{v_{j_i}, v_{j_{i+1}}\}) > B$. Consider a tree T' obtained from T by subdividing the edge $\{v_{j_i}, v_{j_{i+1}}\}$ into two edges $\{v_{j_i}, x_i\}$ and $\{x_i, v_{j_{i+1}}\}$ with weights $(B - \ell(R_i))/2$ and $\omega(\{v_{j_i}, v_{j_{i+1}}\}) - (B - \ell(R_i))/2$, respectively, for each

$i \in \{1, \dots, k-1\}$. (Hence the sum of the two weights of the new edges $\{v_{j_i}, x_i\}$ and $\{x_i, v_{j_i+1}\}$ equals $\omega(\{v_{j_i}, v_{j_i+1}\})$, the weight of the subdivided edge.) Note that the common nodes of T and T' (that is, the nodes of T) are visited by $\text{PDFS}(T)$ and $\text{PDFS}(T')$ in the same order, both $\text{PDFS}(T)$ and $\text{PDFS}(T')$ are B -exploration strategies and the length of each route in $\text{PDFS}(T')$, except for the last one, is of length exactly B . The latter in particular implies

$$|\text{PDFS}(T')| = \left\lceil \frac{\xi(\text{PDFS}(T'))}{B} \right\rceil. \quad (2)$$

Note that

$$\xi(\text{COPT}(T)) = \xi(\text{COPT}(T')) \quad (3)$$

because, informally speaking, a strategy that minimizes the cost never reaches a node of degree two in order to return to previously visited node — thus, in particular, a traversal of $\{v_{j_i}, x_i\}$ is immediately followed by a traversal of $\{x_i, v_{j_i+1}\}$ and vice versa.

From Eq. (2), Theorem 2 applied to T' , Eq. (3) and Observation 3 (used in this order) we conclude that

$$\begin{aligned} |\text{PDFS}(T')| &= \left\lceil \frac{\xi(\text{PDFS}(T'))}{B} \right\rceil \leq 12 \cdot \left\lceil \frac{\xi(\text{COPT}(T'))}{B} \right\rceil = 12 \cdot \left\lceil \frac{\xi(\text{COPT}(T))}{B} \right\rceil \\ &\leq 12 \cdot |\mathcal{R}|, \end{aligned}$$

and hence $|\text{PDFS}(T)| = |\text{PDFS}(T')|$ completes the proof of Theorem 1. \square

5 Bounding the Cost of $\text{PDFS}(T)$

5.1 Additional Notation

When referring to subtrees, we consider them always in the context of their distance from the root. More specifically, we consider the *potential of a node* v , denoted by $\varphi(v)$, to be defined as $\varphi(v) = B/2 - d(r, v)$. If u is the parent of v in T , then we say that u is the *higher endpoint* of $\{u, v\}$ and v is the *lower endpoint* of the edge $\{u, v\}$; we also say that $\{u, v\}$ is a *downward edge* of u . For any subtree T' of T , we define the *potential of T'* , denoted $\varphi(T')$, to be the potential of its root. Then, $2\varphi(T')$ is an upper bound on the total length of any route inside T' . We say that a route *reaches a potential* x in some subtree if it reaches a vertex having potential x . Additionally, for any subtree T' of T , we denote its *weight* to be $\omega(T') = \sum_{e \in E(T')} \omega(e)$, where $E(T')$ is the edge set of T' . In other words, the weight of T is the total weight of its edges. We denote by $T[v]$ the subtree of T rooted at v that contains v and all its descendants, and by $T[e]$ the tree composed of the edge e and $T[v]$ where v is the lower endpoint of e .

We say that a subtree T' of T is *heavy* if $\omega(T') > \varphi(T')$, and otherwise we say that T' is *light*. We extend this terminology to vertices and edges: a vertex v or an

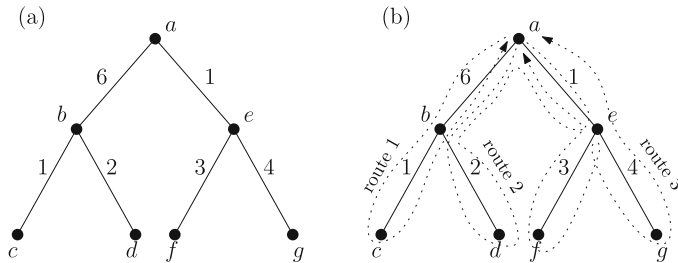


Fig. 3 **a** a tree with a depth first search traversal $(a, b, c, b, d, b, a, e, f, e, g, e, a)$; **b** a B' -adversarial DFS B -exploration S with $B' = 16$ and $B = 20$ has three routes: (a, b, c, b, a) (with length $14 \leq B'$), (a, b, d, b, a, e, a) (with length $18 \leq B$) and (a, e, f, e, g, e, a) (with length $16 \leq B$); $\xi(S) = 14 + 18 + 16 = 48$. Note that two routes (a, b, c, b, d, b, a) and (a, e, f, e, g, e, a) constitute a DFS B -exploration strategy with the cost 34

edge e is heavy if $T[v]$ or $T[e]$ is heavy, respectively. Additionally, by $\text{heavydeg}(v)$ we denote the number of outgoing downward edges of v that are heavy (note that if an edge is heavy, then both its endpoints are heavy as well); we call this parameter the *heavy degree* of T . Observe that if v is any vertex of T and $T[v]$ is heavy, then one route is not enough to cover the entire $T[v]$ in any B -exploration strategy.

5.2 Adversarial DFS-Exploration

When analyzing the cost of $\text{PDFS}(T)$ we will use a recursive approach where the B -exploration of any subtree T' would be defined by taking $2\varphi(T')$ as the maximum route length starting and ending at the root of T' . However the first route that reaches the subtree T' may have performed other explorations before entering T' . Therefore we need to use a slightly generalized DFS B -exploration, called a B' -adversarial DFS B -exploration, denoted by $\text{ADFS}_{B'}(T)$, where the length of the first route is bounded by $B' \leq B$. Moreover, for the simplicity of our arguments we will measure the cost of $\text{ADFS}_{B'}(T)$ by taking the length of the first route to be *exactly* B' and the length of each subsequent route to be *exactly* B . Thus, we artificially increase the cost of $\text{ADFS}_{B'}(T)$ while keeping the number of its routes in this way. Informally speaking, we achieve the following in this way. The fact that some routes may have length strictly smaller than just assumed does not allow us to have any advantage in our arguments. On the other hand, such a ‘rounding’ of the routes’ lengths will simplify some statements, and we will make an appropriate reference to indicate where we explicitly use it. We finally note that bounding the cost of $\text{ADFS}_{B'}(T)$ with such increased cost does trivially give the required bound on the cost of $\text{PDFS}(T)$.

This is formally defined by replacing Eq. (1) for $i = 1$ in condition (ii) in the definition of $\text{PDFS}(T)$ by the following equation (see also Fig. 3):

$$\ell((v_{j_0}, v_{j_0+1}, \dots, v_p)) + d(v_p, r) \leq B'. \quad (4)$$

In other words, the length of the first route is bounded by B' (Eq. (4)) and the lengths of the remaining routes are bounded by B (Eq. (1) for $i > 1$).

For a given tree T , we define an *adversarial DFS exploration* of T , denoted by $\text{ADFS}(T)$, and defined to be an exploration strategy $\text{ADFS}_{B'}(T)$ that maximizes the cost, across all possible values of B' :

$$\xi(\text{ADFS}(T)) = \max_{0 \leq B' \leq 2\varphi(T)} \xi(\text{ADFS}_{B'}(T)). \quad (5)$$

In the following analysis, it will be convenient for us to use arguments that rely on the fact that B' , for our purposes, takes only one of the finite values from $[0, 2\varphi(T)]$. This is due to the above comment, namely, for $T[r]$, where r is the root of T , we have that $B' = B$ and for any other subtree $T[v]$, the values of B' interesting for us depend on the prefix of the route that starts at the root of T and reaches v . Thus, a simple inductive argument allows us to conclude that the value of B' depends on all possible DFS B -exploration strategies of T (the number of those is finite). Hence, we denote by \mathcal{B} the finite set that consists all values $B' = B - x - d(r, v)$ such that there exists a route in T of length x that starts at the root of T and ends at v . Thus, we can restate (5):

$$\xi(\text{ADFS}(T)) = \max_{B' \in \mathcal{B}} \xi(\text{ADFS}_{B'}(T)).$$

Note that it follows from the definition that

$$\xi(\text{ADFS}(T)) \geq \xi(\text{PDFS}(T)).$$

Intuitively, if v is any node of the tree T , then $\text{ADFS}(T[v])$ is the worst case scenario of how a DFS B -exploration may perform in $T[v]$ in terms of the cost; this worst case is understood as considering the worst possible ending point of the route that (in the entire tree T) precedes the considered strategy $\text{ADFS}(T[v])$.

Lemma 4 *If v is any node of T and e_1, e_2, \dots, e_k are all downward edges of v , then*

$$\xi(\text{COPT}(T[v])) = \sum_{1 \leq i \leq k} \xi(\text{COPT}(T[e_i])), \quad (6)$$

$$\xi(\text{ADFS}(T[v])) \leq \sum_{1 \leq i \leq k} \xi(\text{ADFS}(T[e_i])). \quad (7)$$

Proof Informally, equality in (6) for an optimal solution follows from the fact that $\text{COPT}(T[v])$ has the freedom to pick the length of each route to be an arbitrary number in \mathcal{B} . Any strategies for $T[e_1], T[e_2], \dots, T[e_k]$ can be translated into strategy for $T[v]$: the latter one is constructed by simply concatenating the former exploration strategies. Similarly, if one takes an exploration strategy $\text{COPT}(T[v])$, then one can assume without affecting its cost that each of its routes has only two occurrences of the root: it is the first and last vertex of the route. But then, such a strategy $\text{COPT}(T[v])$ can be partitioned into the corresponding strategies for the trees $T[e_1], T[e_2], \dots, T[e_k]$.

We now prove (7). Consider an exploration strategy $\text{ADFS}(T[v])$. Each route of this strategy is of length at most $2\varphi(T)$. Obtain an exploration strategy \mathcal{S} by partitioning each route in $\text{ADFS}(T[v])$ in such a way that the concatenation of all routes in \mathcal{S}

equals the concatenation of all routes in $\text{ADFS}(T[v])$ and no route in \mathcal{S} has v as an internal vertex. (Thus, each route of \mathcal{S} starts and ends at v .) Note that $\xi(\mathcal{S}) = \xi(\text{ADFS}(T[v]))$. Now, \mathcal{S} can be partitioned into $\mathcal{S}_1, \dots, \mathcal{S}_k$ such that \mathcal{S}_i is a b'_i -adversarial DFS exploration strategy of $T[e_i]$ for some $b'_i \leq 2\varphi(T[v])$, i.e., $\mathcal{S}_i = \text{ADFS}_{b'_i}(T[e_i])$, for each $i \in \{1, \dots, k\}$ and the concatenation of $\mathcal{S}_1, \dots, \mathcal{S}_k$ gives \mathcal{S} . Thus,

$$\xi(\text{ADFS}(T[v])) = \xi(\mathcal{S}) = \sum_{i=1}^k \xi(\text{ADFS}_{b'_i}(T[e_i])).$$

To conclude the proof, observe that by the definition of adversarial exploration

$$\xi(\text{ADFS}_{b'_i}(T[e_i])) \leq \xi(\text{ADFS}(T[e_i])), \quad i \in \{1, \dots, k\}.$$

□

The proof of Theorem 2 will follow from the following two results: Theorem 5 and Lemma 6, and the fact that $\xi(\text{PDFS}(T)) \leq \xi(\text{ADFS}(T))$. In order to prove Theorem 5 we will first define a special class of heavy trees called *Skinny Tree* which has the following property.

(ST) Skinny Tree Property: *If any vertex v of T has heavy degree equal to one, then consider the longest path in T that connects v to such a descendant r' that each internal vertex of the u - r' path has heavy degree equal to 1 (i.e., each edge of the path is heavy). We then require that each vertex of this path, except for r' , has at most one light edge incident to it.*

For the statement of Theorem 5 we need a parameter that we denote by $\vartheta(T)$, for an arbitrary tree T . The reason for introducing it is technical and can be intuitively explained as follows. In Theorem 5 we essentially need a strict inequality for trees satisfying property (ST), e.g., in case of (i) we need $\xi(\text{ADFS}(T)) < 12 \cdot \xi(\text{COPT}(T)) - 10 \cdot \varphi(T)$. This separation between the left and right hand sides of this inequality is needed for skinny trees because of the transformation from an arbitrary tree to a tree satisfying (ST) in Sect. 6. Thus, Theorem 5 is proved for skinny trees first (Sect. 5.3). Then, in the end we need it to work for general trees, but here it is enough to have a weak inequality. This distinction between weak and strict inequality for these two types of trees dictates the definition of $\vartheta(T)$: if T is skinny, then $\vartheta(T)$ is the minimum weight taken over all edges incident to the leaves of T , and if T is not skinny, then let $\vartheta(T) = 0$. The exact motivation for such a definition for a skinny tree will become clear in Sect. 6.

Theorem 5 *If T is heavy, and r is its root, then:*

- (i) *if $\text{heavydeg}(r) = 1$, then $\xi(\text{ADFS}(T)) \leq 12 \cdot \xi(\text{COPT}(T)) - 10 \cdot \varphi(T) - \vartheta(T)$,*
- (ii) *if $\text{heavydeg}(r) \neq 1$, then $\xi(\text{ADFS}(T)) \leq 12 \cdot \xi(\text{COPT}(T)) - 20 \cdot \varphi(T) - \vartheta(T)$.*

We will use the following technical lemma which states that it is possible to rearrange any tree to have the above property:

Lemma 6 For any tree T , there exists $\varepsilon > 0$ and a corresponding tree T_ε such that (i) T_ε satisfies Property (ST), and (ii) if Theorem 5 holds for T_ε , then Theorem 5 holds for T .

The proof of Lemma 6 is postponed to Sect. 6. Due to the above result, we can now focus on proving Theorem 5 for any tree T with the above-mentioned property in the rest of Sect. 5.

5.3 Proof of Theorem 5 for Skinny Trees

We start with a technical lemma.

Lemma 7 For any tree T , if T is light, then $\xi(\text{ADFS}(T)) \leq 2 \cdot \xi(\text{COPT}(T)) - \vartheta(T)$.

Proof If T is light, then observe that $\text{ADFS}(T)$ either consists of one route, in which case $\xi(\text{ADFS}(T)) = \xi(\text{COPT}(T))$, or it contains at least two routes but then the second route, having length up to $2\varphi(T)$, will explore all remaining vertices of T since $\varphi(T) \geq \omega(T)$ holds for a light tree. Thus, in the latter case $\text{ADFS}(T)$ has exactly two routes, one of them omitting one edge incident to a leaf and thus having length at most $2\omega(T) - \vartheta(T)$, which gives $\xi(\text{ADFS}(T)) \leq 4\omega(T) - \vartheta(T) \leq 2 \cdot \xi(\text{COPT}(T)) - \vartheta(T)$. \square

To prove Theorem 5, we will proceed by induction on the number of heavy edges in a tree. This is a valid approach since the parent of a heavy node is also heavy. Moreover, the inductive argument will be applied correctly because if T is skinny and v is any vertex of T , then $T[v]$ is also skinny. We note that the theorem is proved for skinny trees and thus we keep in mind that $\vartheta(T)$ is the minimum weight over all edges incident to the leaves of T in this section.

For the base case consider T with no heavy edges. In particular we have that $\text{heavydeg}(r) = 0$. Denote downward edges at r by e_1, e_2, \dots, e_l . For each $i \in \{1, \dots, l\}$, $T[e_i]$ is light and hence by Lemma 7, $\xi(\text{ADFS}(T[e_i])) \leq 2 \cdot \xi(\text{COPT}(T[e_i])) - \vartheta(T)$. Thus in particular, by (6) and (7) and $\xi(\text{COPT}(T)) \geq 2\omega(T)$,

$$\begin{aligned} \xi(\text{ADFS}(T)) &\leq 2\xi(\text{COPT}(T)) - \vartheta(T) \\ &\leq 12\xi(\text{COPT}(T)) - 20\omega(T) - \vartheta(T) \\ &\leq 12\xi(\text{COPT}(T)) - 20\varphi(T) - \vartheta(T). \end{aligned}$$

For the induction step, we assume that Theorem 5 holds for all heavy proper subtrees of T . In what follows we consider two cases: when $\text{heavydeg}(r) > 1$ and $\text{heavydeg}(r) = 1$.

Case of $\text{heavydeg}(r) > 1$

Let e_1, \dots, e_h be the heavy downward edges at r and e'_1, \dots, e'_l be the light downward edges at r . By induction hypothesis (and precisely Theorem 5(i)) we obtain

$$\xi(\text{ADFS}(T[e_i])) \leq 12 \cdot \xi(\text{COPT}(T[e_i])) - 10\varphi(T[e_i]) - \vartheta(T[e_i])$$

for each $i \in \{1, \dots, h\}$. Then, by (7) of Lemma 4, by Lemma 7, and by $\varphi(T) = \varphi(T[e_i])$ (used in this order):

$$\begin{aligned} \xi(\text{ADFS}(T)) &\leq \sum_{i=1}^h \xi(\text{ADFS}(T[e_i])) + \sum_{i=1}^l \xi(\text{ADFS}(T[e'_i])) \\ &\leq \sum_{i=1}^h (12\xi(\text{COPT}(T[e_i])) - 10\varphi(T[e_i]) - \vartheta(T[e_i])) \\ &\quad + \sum_{i=1}^l 2\xi(\text{COPT}(T[e'_i])) \\ &\leq 12 \left(\sum_{i=1}^h \xi(\text{COPT}(T[e_i])) + \sum_{i=1}^l \xi(\text{COPT}(T[e'_i])) \right) \\ &\quad - 10\varphi(T) \cdot h - \vartheta(T) \\ &\leq 12\xi(\text{COPT}(T)) - 20\varphi(T) - \vartheta(T). \end{aligned}$$

The last inequality is due to (6) and $h = \text{heavydeg}(r) \geq 2$.

Case of $\text{heavydeg}(r) = 1$

Let r' be the closest descendant of r in T that is heavy and satisfies $\text{heavydeg}(r') \neq 1$. Note that such a vertex always exists and r' is unique. Let P denote the path connecting r to r' . Additionally, we denote by e_1, e_2, \dots, e_l all light edges incident to the vertices in $V(P) \setminus \{r'\}$ in the *non-decreasing* order of their potentials. (We remark here that the subtree rooted at r' has been covered by the base case of the induction and by the case when the heavy degree is greater than one.) Denote $\varphi_i = \varphi(T[e_i])$, $i \in \{1, \dots, l\}$. Due to Lemma 6 (and more precisely by the fact that thanks to Lemma 6 we assume that in the tree T all edges e_1, \dots, e_l have pairwise different potentials) we have:

$$\varphi(r) \geq \varphi_l > \dots > \varphi_2 > \varphi_1 > \varphi(r') = \varphi_0.$$

See Fig. 4(a) that illustrates the path P and placements of the edges e_i and the corresponding potentials.

We take for brevity

$$w_i = \omega(T[e_i]) = \frac{1}{2}\xi(\text{COPT}(T[e_i])) \quad \text{for } i \in \{1, \dots, l\}, \quad w_0 = \frac{1}{2}\xi(\text{COPT}(T[r'])).$$

Let c be the number of routes in $\text{COPT}(T)$, and we denote by x_i the lowest potential i -th route reached on the path P (we ignore potentials it reached in subtrees — see Fig. 4(b)), where the routes are without loss of generality ordered so that $x_1 \leq x_2 \leq \dots \leq x_c$.

Consider $j \in \{1, \dots, c\}$. Informally speaking, the first j routes of $\text{COPT}(T)$ need to cover all subtrees $T[e_i]$ such that the potential of the higher endpoint of e_i is *strictly*

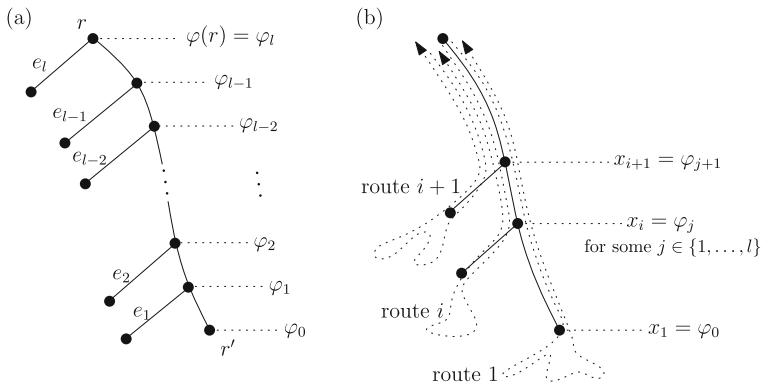


Fig. 4 **a** the path P (although it may contain vertices of degree two, they are trivial from the point of view of the analysis, thus omitted in the picture); **b** illustration of x_i 's. Note that the first route (and possibly arbitrarily many more routes) always reaches r'

smaller than x_j ; otherwise some vertices would not be visited by $\text{COPT}(T)$. The total weight of these subtrees is $\sum_{i:\varphi_i < x_j} w_i$. Observe, that the total length of all parts of an i -th route that do not belong to the path P is at most $2x_i$. Thus, the total weight of the above-mentioned subtrees satisfies

$$\sum_{i:\varphi_i < x_j} w_i \leq x_1 + \cdots + x_{j-1} \quad \text{for each } j \in \{1, \dots, c+1\}, \quad (8)$$

where we denote $x_{c+1} = +\infty$ for the sake of simplicity.

We are now interested in bounding the cost of $\text{ADFS}(T)$ on the path P with respect to $\sum_{i=1}^c 2(\varphi(r) - x_i)$, that is, with respect to the cost of $\text{COPT}(T)$ on the path P . To do this, we start by comparing x_1, \dots, x_c with y_1, \dots, y_d chosen by an appropriate greedy procedure:

$$y_j = \min \left\{ y : \sum_{i:\varphi_i \leq y} w_i > y_1 + \cdots + y_{j-1} \right\}, \quad (9)$$

$$d = \min \{ i : y_1 + \cdots + y_i \geq w_0 + w_1 + \cdots + w_l \},$$

in other words, assigning y_j to be the first value where (8) is violated given only y_1, \dots, y_{j-1} . (Notice that from the definition we have that always $y_j \in \{\varphi_0, \varphi_1, \dots, \varphi_l\}$. Moreover, if $y_j > \varphi_0$, then $y_{j+1} > y_j$.) We obtain the following lemma which says that, across all sequences satisfying (8), y_i takes maximal values:

Lemma 8 *It holds that $d \leq c$ and $y_j \geq x_j$ for each $j \in \{1, \dots, d\}$.*

Proof We proceed by induction on j .

For the inductive base, we have $x_1 = y_1 = \varphi(r')$. For the inductive step, assume the claim holds for all indices smaller than j . Suppose for a contradiction that $x_j > y_j$. Then,

$$\sum_{i:\varphi_i \leq y_j} w_i \leq \sum_{i:\varphi_i < x_j} w_i$$

and by (8) and the inductive assumption applied for all indices smaller than j ,

$$\sum_{i:\varphi_i < x_j} w_i \leq y_1 + \dots + y_{j-1}.$$

These two inequalities give a contradiction with (9). \square

Observe that $\text{ADFS}(T)$ first traverses (in that order) some subset of light subtrees $T[e_i]$, whose indices we denote by $H \subseteq \{1, \dots, l\}$, in a decreasing order of their indices. The above routes, none of which contains r' , will form the *first part of* $\text{ADFS}(T)$. Then, all vertices of $T[r']$ are visited (to those routes of $\text{ADFS}(T)$ we refer as the *second part of* $\text{ADFS}(T)$) and following that, remaining light subtrees $T[e_i]$ for $i \in H' = \{1, \dots, l\} \setminus H$, in an increasing order of their indices (*third part of* $\text{ADFS}(T)$). Note that there may exist a route that has a non-empty intersection with a tree $T[e_i]$, $i \in H \cup H'$, and also contains r' — this route belongs by definition to the second part of $\text{ADFS}(T)$.

Denote by z_1, \dots, z_p the lowest potentials reached by subsequent routes in $\text{ADFS}(T)$ on the path P (ignoring as before the potentials they reach in subtrees), only in the first part of $\text{ADFS}(T)$ in the reversed order of entering T :

$$\varphi(r) \geq z_p \geq \dots \geq z_1 > \varphi(r'). \quad (10)$$

We note that each route in the first part of $\text{ADFS}(T)$ visits subtrees with a continuous segment of indices from H , that is $T[e_i]$'s for $i \in H \cap \{i_1, \dots, i_2\}$ for some integers i_1, i_2 . Due to the weight of a light tree, its vertices belong to at most two different routes.

Lemma 9 *If for some i, j there is $z_i \geq y_j$, then $z_{i+2} \geq y_{j+1}$.*

Proof If $y_{j+1} = y_j$ then the claim follows immediately from the fact that, by (10), $z_{i+1} \geq z_i \geq y_j = y_{j+1}$. Similarly, if $z_i \geq y_{j+1}$ then by (10) we have $z_{i+2} \geq z_i \geq y_{j+1}$. Thus, assume that $y_{j+1} \geq z_i$ and $y_{j+1} > y_j$.

Let $a < b$ be indices such that $y_j = \varphi_a$ and $y_{j+1} = \varphi_b$. We have from the way y_j is selected in (9):

$$w_0 + \dots + w_b > y_1 + \dots + y_j \geq w_0 + \dots + w_{b-1},$$

$$w_0 + \dots + w_a > y_1 + \dots + y_{j-1} \geq w_0 + \dots + w_{a-1}.$$

Thus,

$$y_j > w_{a+1} + \dots + w_{b-1}.$$

This inequality, informally speaking, certifies that the total weight of all subtrees $T[e_s]$ with $s \in \{a+1, \dots, b-1\}$ is at most y_j . By assumption $z_i \geq y_j$. Thus, a route that traverses all subtrees $T[e_{a+1}], \dots, T[e_{b-1}]$ has length bounded by $2y_j$ and

hence its length is at most $2z_i$. By the definition of $T[e_a]$, $w_a \leq \varphi_a = y_j$. Thus, one extra route of $\text{ADFS}(T)$ is guaranteed to cover $T[e_a]$. Thus, two subsequent routes of $\text{ADFS}(T)$ are enough to fully cover the subtrees $T[e_a], \dots, T[e_{b-1}]$. Particularly, the two mentioned routes are the ones that reach potentials z_i and z_{i+1} . This implies that the route that precedes these ones, i.e., the route that reaches the potential z_{i+2} does not reach potentials smaller than φ_b . Thus, $z_{i+2} \geq \varphi_b = y_{j+1}$ as required in the lemma. \square

We are now ready to bound the total cost of $\text{ADFS}(T)$ with relation to $\text{COPT}(T)$. The cost of $\text{COPT}(T)$ can be decomposed:

$$\xi(\text{COPT}(T)) = O_{\text{light}} + O_{\text{deep}} + O_{\text{path}} + O_{\text{flat}}, \quad (11)$$

where:

- O_{light} — is the cost restricted to light subtrees $T[e_i]$ with $i \in \{1, \dots, l\}$,
- O_{deep} — is the cost restricted to the subtree $T[r']$,
- O_{path} — is the cost restricted to the path P and the routes that do not contain r' , and
- O_{flat} — is the cost restricted to the path P and routes that do contain r' .

Similarly, we express the cost of $\text{ADFS}(T)$ as a sum:

$$\xi(\text{ADFS}(T)) = D_{\text{light}} + D_{\text{deep}} + D_{\text{desc}} + D_{\text{flat}} + D_{\text{asc}}, \quad (12)$$

where

- D_{light} — is the cost of $\text{ADFS}(T)$ restricted to light subtrees $T[e_i]$ with $i \in \{1, \dots, l\}$,
- D_{deep} — is the cost restricted to the subtree $T[r']$,
- D_{desc} — is the cost restricted to the path P in the first part of $\text{ADFS}(T)$,
- D_{flat} — is the cost restricted to P and the routes that contain r' (i.e., the second part of $\text{ADFS}(T)$), and
- D_{asc} — is the cost restricted to the path P in the third part of $\text{ADFS}(T)$.

By Lemma 7, $\xi(\text{ADFS}(T[e_i])) \leq 2\xi(\text{COPT}(T[e_i])) - \vartheta(T[e_i])$ for each $i \in \{1, \dots, l\}$ and therefore

$$D_{\text{light}} \leq 2 \cdot O_{\text{light}} - \vartheta(T). \quad (13)$$

Denote by s the smallest index such that $y_{s+1} > \varphi_0 = y_s$ and let j_1, \dots, j_q , $j_1 < j_2 < \dots < j_q$, be the potentials reached by subsequent routes in $\text{ADFS}(T)$. Since $z_{j_1} \geq y_s$, by iteratively applying Lemma 9 we obtain that

$$z_{j_i} \geq y_{s+[i/2]-1} \text{ for each } i \in \{1, \dots, q\}.$$

Therefore, by Lemma 8,

$$\begin{aligned}
 D_{desc} &= 2 \sum_{i=1}^q (\varphi(T) - z_{j_i}) = 2(\varphi(T) - \varphi_0) + 2 \sum_{i=2}^q (\varphi(T) - z_{j_i}) \\
 &\leq 2\omega(P) + 2 \sum_{i=2}^q (\varphi(T) - y_{s+\lceil i/2 \rceil - 1}) \\
 &\leq 2\omega(P) + 4 \sum_{i=2}^q (\varphi(T) - y_{s+i-1}) \\
 &\leq 2\omega(P) + 4 \sum_{i=2}^q (\varphi(T) - x_{s+i-1}) \leq 2\omega(P) + 2O_{path}.
 \end{aligned} \tag{14}$$

By an analogous analysis, the same bound holds for the third part of ADFS(T):

$$D_{asc} \leq 2\omega(P) + 2O_{path}. \tag{15}$$

By the inductive assumption we have

$$D_{deep} < 12 \cdot O_{deep} - 20\varphi(r'). \tag{16}$$

We also get the following bounds by analyzing how much each particular route can overlap with $T[r']$. The first one follows from an observation that each route having a non-empty intersection with $T[r']$ may have length restricted to $T[r']$ at most $2\varphi(r')$. Thus, there exist at least $O_{deep}/(2\varphi(r'))$ such routes in $\text{COPT}(T)$ intersecting $T[r']$ and each such route contributes at least $2\omega(P)$ to O_{flat} . Hence,

$$O_{flat} \geq \frac{\omega(P)}{\varphi(r')} \cdot O_{deep}. \tag{17}$$

As for an upper bound, there exist at most $\lceil D_{deep}/(2\varphi(r')) \rceil + 1$ routes in $\text{ADFS}(T)$ that contain r' (note that the first such route may include no other vertices except for r' from $T[r']$ and thus the additive factor of “+1”). Also note that we use here the assumption that the length of each route in $\text{ADFS}(T)$, except for the first one, is exactly $\varphi(r')$. This assumption comes from the definition of $\text{ADFS}(T)$. Thus,

$$D_{flat} \leq 2\omega(P) \cdot \lceil D_{deep}/(2\varphi(r')) \rceil + 1 \leq 4\omega(P) + \omega(P) \cdot D_{deep}/\varphi(r'). \tag{18}$$

Equations (18), (16) and (17), used in that order, give us

$$\begin{aligned}
 D_{deep} + D_{flat} &\leq 4\omega(P) + \left(\frac{\omega(P)}{\varphi(r')} + 1\right) D_{deep} \\
 &\leq 4\omega(P) + \left(\frac{\omega(P)}{\varphi(r')} + 1\right) (12 \cdot O_{deep} - 20\varphi(r')) \\
 &\leq 4\omega(P) - 20(\omega(P) + \varphi(r')) + 12 \cdot O_{deep} \left(\frac{\omega(P)}{\varphi(r')} + 1\right) \\
 &\leq 12O_{deep} + 12O_{flat} - 16\omega(P) - 20\varphi(r').
 \end{aligned}$$

Then, by (13), (14) and (15) we have

$$D_{light} + D_{asc} + D_{desc} \leq 2O_{light} - \vartheta(T) + 4\omega(P) + 4O_{path}.$$

The last two inequalities, (11), (12) and $\varphi(r) = \omega(P) + \varphi(r')$ finally give

$$\begin{aligned}
 \xi(\text{ADFS}(T)) &\leq 12O_{deep} + 12O_{flat} + 2O_{light} + 4O_{path} \\
 &\quad - 12\omega(P) - 20\varphi(r') - \vartheta(T) \\
 &\leq 12\xi(\text{COPT}(T)) - 10\varphi(r) - \vartheta(T),
 \end{aligned}$$

which completes the inductive proof of Theorem 5.

6 Tree Rearrangement

This section is devoted to proving Lemma 6. We start with an informal description providing a high level intuition that gives an overview of this section. Our first step (Sect. 6.2) is to construct a tree T_ε , $\varepsilon \in \mathbb{R}^+$, based on T such that T_ε satisfies property (ST). We will also need that T_ε ‘resembles’ T in the following way: an exploration strategy is valid for T if and only if a ‘very similar’ strategy is valid for T_ε . To simplify the statements considerably, it will be convenient to encode strategies in an uniform way so that we can apply the same strategy for both trees, without going into the details of tedious but straightforward conversions between strategy for T and strategy for T_ε . We thus define (Sect. 6.1) a collection \mathcal{C} of all possible strategies (including adversarial ones and those that are not feasible for either T or T_ε because they contain routes that are too long or do not visit all vertices). Then in Sect. 6.3 we select the right value of ε . The value of ε and the construction of T_ε will ensure that a strategy in \mathcal{C} is valid for T if and only if it is valid for T_ε . We then finally provide the main result of this section (Lemma 6), which states that, again thanks to the choice of ε , if Theorem 5 holds for T_ε , then it holds for T , thus justifying our approach in Sect. 5, where we restricted the analysis only to trees satisfying property (ST).

6.1 Finding the Right Value of ε

We define a *potential route* as a pair $R' = (L, v)$, where $L = (l_1, \dots, l_p)$ is a sequence of leaves and $v \in V(T_\varepsilon)$. Then, R' translates to a route R in T as a concatenation of the following paths (in this order): the path from r to l_1 , the path from l_i to l_{i+1} , $i = 1, \dots, p-1$, the path from l_p to the closest ancestor x of v that belongs to $V(T)$ and finally the path from x to r . The length of R is

$$\ell(R) = d(r, l_1) + d(l_p, x) + d(x, r) + \sum_{1 \leq i < p} d(l_i, l_{i+1}).$$

R' translates to a route in T_ε in the same way, except that we take $x = v$, i.e., v is not replaced by the ancestor. Then, a *potential strategy* is a sequence consisting of at most j potential routes, $j \in \{1, \dots, p\}$, where p is the number of leaves in T .

Note that a potential strategy may not translate to a valid B -exploration strategy for T or T_ε because some nodes may not be explored and some routes may be too long. A potential strategy is *feasible* for T (respectively T_ε) if it translates to a valid B -exploration strategy for T (respectively T_ε). We denote by \mathcal{C} the collection of all potential strategies. Clearly, the size of \mathcal{C} is finite.

We conclude with the following:

observation 10 For any route R that may appear in $\text{ADFS}(T)$, $\text{COPT}(T)$, $\text{ADFS}(T_\varepsilon)$ and $\text{COPT}(T_\varepsilon)$ there exists a potential route that translates to R . \square

6.2 The Construction of T_ε

We now construct the tree $T_\varepsilon = (V(T) \cup X, E(T_\varepsilon), \omega')$ based on $T = (V(T), E(T), \omega)$ and the construction depends on a parameter $\varepsilon > 0$ that will be fixed later. We now impose only a condition on ε that is needed for the construction itself to be valid:

$$\varepsilon < \min\{\omega(e) \mid e \in E(T)\}.$$

Select an arbitrary vertex u in T with $\text{heavydeg}(u) = 1$ (denote by $\{u, v\}$ the heavy downward edge at u) and $d > 1$ light downward edges e_1, \dots, e_d at v . Subdivide the edge $e = \{u, v\}$ (see Fig. 5 for an illustration) by replacing it by a path $P = (u = v_0, v_1, \dots, v_{d-1}, v_d = v)$ with the following edge lengths: the first $d-1$ edges have length $\varepsilon/(d-1)$, i.e., $\omega'(v_i, v_{i+1}) = \varepsilon/(d-1)$ for each $i \in \{0, \dots, d-2\}$, and for the last edge we set $\omega'(\{v_{d-1}, v\}) = \omega(\{u, v\}) - \varepsilon$. (Note that this preserves the distance between u and v .) Then, the weight of each edge e_i decreases by ε , $\omega'(e_i) = \omega(e_i) - \varepsilon$ and the higher endpoint of e_i in T_ε becomes v_{i-1} , $i \in \{1, \dots, d\}$. (Note that this ensures that the distance between u and the lower endpoint of e_i or the distance between two children of u is not greater in T_ε than in T .) This construction allows us to assume (by permuting the edges e_1, \dots, e_d appropriately) that the DFS traversal of T_ε visits the edges e and e_1, \dots, e_d in the same order both in T and in T_ε , ensuring that Condition (P2), which is defined below, is satisfied.

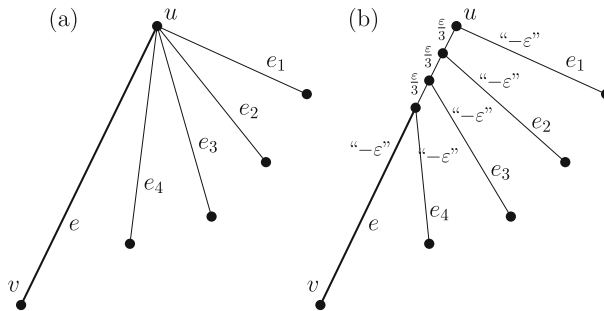


Fig. 5 **a** the node u of T with $\text{heavydeg}(u) = 1$ and light downward edges e_1, \dots, e_d , $d = 4$; **b** the corresponding edges in T_ε (the “ $-\varepsilon$ ” indicated that the weight of the edge is ε smaller than the weight of the corresponding edge in T)

Since the vertex v is selected arbitrarily, we repeat the above modification for each such vertex v obtaining the final tree T_ε .

We will require the following conditions to be satisfied:

- (P1) there are at most two downward edges at each vertex in T_ε with $\text{heavydeg}(v) = 1$,
- (P2) there exists a DFS traversal of T_ε that visits the vertices in $V(T)$ in the same order as the DFS traversal that we have fixed for T in this work,
- (P3) $\mathcal{S} \in \mathcal{C}$ is feasible for T if and only if \mathcal{S} is feasible for T_ε .

Property (P3) will be proved in Lemma 13 and we now note:

observation 11 For each $\varepsilon > 0$, the tree T_ε satisfies Conditions (P1) and (P2). \square

observation 12 The strategies $\text{ADFS}(T)$ and $\text{ADFS}(T_\varepsilon)$ visit the nodes in $V(T)$ in the same order. \square

6.3 The Analysis of T_ε

In this section we argue that the construction of T_ε ‘preserves’ the problem: the minimum costs of adversarial DFS explorations of both T and T_ε , as well as $\text{COPT}(T)$ and $\text{COPT}(T_\varepsilon)$ remain close to each other for ε small enough. Intuitively speaking, this follows from a ‘continuity argument’ formalized in the remaining part of this section.

For the tree T_ε we define an interval denoted by $\mathcal{I} = (0, y)$, $y < \min\{\omega(e) \mid e \in E(T)\}$, such that for each $\varepsilon \in (0, y)$, Condition (P3) holds for T_ε . We now prove that this interval is well defined, that is, $y > 0$. (Note that for $\varepsilon = 0$, T_ε and T are the same.)

Lemma 13 It holds $\mathcal{I} \neq \emptyset$.

Proof First we argue that there exists $y > 0$ such that T_y fulfills Condition (P3). We select y based on the tree T and the collection \mathcal{C} . Consider any $B' \in \mathcal{B}$. The number of potential strategies in \mathcal{C} is finite, and hence the number of potential strategies in \mathcal{C} that do not translate to feasible ones for T (denote subset of those by \mathcal{C}') is also finite.

For each $S \in \mathcal{C}'$, define its *deficiency* $x(S)$ as follows: the $x(S)$ is the maximum value such that either the length of the first route in S is $B' + x(S)$ or the length of some other route in S in T is $B + x(S)$. Intuitively, S does not translate to a feasible B' -adversarial B -exploration strategy for T because one of its routes exceeds the allowed length by $x(S)$ and no route exceeds it by more than $x(S)$. Take

$$y := \frac{1}{2n(m+1)} \cdot \min \{ \min \{ \omega(e) \mid e \in E(T) \}, \min \{ x(S') \mid S' \in \mathcal{C}' \} \},$$

where m is the number edges in T . Since by definition, $x(S) > 0$ for each $S \in \mathcal{C}'$, we obtain that $y > 0$. Also, no route in any $S \in \mathcal{C}'$ traverses an edge more than $2n$ times and hence the length of any route of S in T_y decreases by at most $2ynm$ with respect to its length in T . This implies that some route R of S has length in T_y at least

$$\tilde{B} + x(S) - 2ynm \geq \tilde{B} + 2yn(m+1) - 2ynm > \tilde{B}, \quad (19)$$

where we take $\tilde{B} = B'$ if R is the first route in S and $\tilde{B} = B$ otherwise. Therefore, S remains unfeasible in T_y . Since an exploration strategy that is feasible in T remains feasible in T_y (recall that the length of each route is smaller in T_y than in T), we have that Condition (P3) holds for T_y .

Finally, observe that substituting y by any value ε smaller than y in the left hand side of (19) keeps this equation true. Therefore, Condition (P3) is satisfied by T_ε for each $\varepsilon \in (0, y)$, which completes the proof. \square

Before we state the main lemma of this section, we prove these technical bounds:

Lemma 14 *For each $\varepsilon \in \mathcal{I}$ it holds:*

- (i) $\xi(\text{ADFS}(T)) \leq \xi(\text{ADFS}(T_\varepsilon)) + 4\varepsilon n^2$,
- (ii) $\varphi(T) \leq \varphi(T_\varepsilon) + 2n\varepsilon$, and
- (iii) $\xi(\text{COPT}(T_\varepsilon)) \leq \xi(\text{COPT}(T))$.

Proof By Observation 12, both $\text{ADFS}(T)$ and $\text{ADFS}(T_\varepsilon)$ visit the leaves of both trees in the same order. Consider any edge $\{u, v\}$ in T such that v is its lower endpoint. We have that there exists a *corresponding* edge $\{u', v\}$ in T_ε . Moreover, $\omega(\{u, v\}) \leq \omega'(\{u', v\}) + \varepsilon$. Due to Condition (P3), $\text{ADFS}(T)$ is valid because $\text{ADFS}(T_\varepsilon)$ is valid. This implies that for each edge traversal in $\text{ADFS}(T_\varepsilon)$, a traversal of the corresponding edge occurs in $\text{ADFS}(T)$. Thus, if t is the number of edge traversals in $\text{ADFS}(T)$, then

$$\xi(\text{ADFS}(T)) \leq \xi(\text{ADFS}(T_\varepsilon)) + t\varepsilon \leq \xi(\text{ADFS}(T_\varepsilon)) + 4n^2\varepsilon.$$

The latter inequality follows from bounding each route in $\text{ADFS}(T_\varepsilon)$ to have at most $2n$ edges, bounding the number of routes by n and observing that an edge is traversed at most twice in each route.

Property (ii) is a direct consequence of the construction of T_ε : any path in T that connects a leaf to the root corresponds to a path in T_ε that connects the same common nodes and contains all common edges of the original path, and some edges that belong only to T_ε . There are at most n of the former edges, each of which differs in length

in T and T_ε by at most ε , and there are at most n of the latter edges, each of which has weight at most ε . This gives an upper bound of $2n\varepsilon$ for the difference between the length of an arbitrary leaf-root path of T and T_ε , and this bounds the difference in their potentials.

By construction of T_ε , the distance between any common nodes $u \in V(T)$ and $v \in V(T)$ is not greater in T_ε than in T , which immediately gives (iii). \square

Proof of Lemma 6 Proof of Lemma 6 We will analyze condition (i) in Theorem 5 and the proof for (ii) is analogous as we note at the end of the proof.

We may assume (by taking ε small enough, i.e., $\varepsilon < \vartheta(T)/2$) that all trees T_ε we consider satisfy $\vartheta(T_\varepsilon) \geq \vartheta(T)/2$. Thus, $\vartheta(T_\varepsilon)$ is in this sense independent of however small is the chosen value of $\varepsilon \in \mathcal{I}$. Then by Theorem 5(i):

$$\xi(\text{ADFS}(T_\varepsilon)) \leq 12 \cdot \xi(\text{COPT}(T_\varepsilon)) - 10 \cdot \varphi(T_\varepsilon) - \vartheta(T_\varepsilon).$$

By Lemma 13, there exists $\varepsilon \in \mathcal{I}$ such that $\varepsilon > 0$ and

$$\varepsilon < \frac{\vartheta(T)}{12n^2},$$

such that T_ε satisfies Condition (P3). Then, by Lemma 14 (in particular, (i) of Lemma 14 is used to obtain the first inequality below and (ii) and (iii) are used to obtain the fourth inequality below) we get:

$$\begin{aligned} \xi(\text{ADFS}(T)) &\leq \xi(\text{ADFS}(T_\varepsilon)) + 4\varepsilon n^2 \\ &\leq 12 \cdot \xi(\text{COPT}(T_\varepsilon)) - 10 \cdot \varphi(T_\varepsilon) - \vartheta(T_\varepsilon) + 4\varepsilon n^2 \\ &\leq 12 \cdot \xi(\text{COPT}(T_\varepsilon)) - 10 \cdot \varphi(T_\varepsilon) - \frac{\vartheta(T)}{2} + 4\varepsilon n^2 \\ &\leq 12 \cdot \xi(\text{COPT}(T)) - 10 \cdot \varphi(T) + 2n\varepsilon - \frac{\vartheta(T)}{2} + 4\varepsilon n^2 \\ &\leq 12 \cdot \xi(\text{COPT}(T)) - 10 \cdot \varphi(T). \end{aligned}$$

Since $\vartheta(T) = 0$ by definition, this completes the proof. We can conduct the same argument for Theorem 5(ii) with the same value of ε . Hence we obtain that Theorem 5 holds for T . \square

7 Conclusions and Open Problems

We studied the problem of exploring a tree in a piecemeal fashion using multiple routes, each restricted to a length of at most B . Our strategy $\text{PDFS}(T)$ achieves the asymptotically minimum number of routes and also minimizes the total cost of exploration up to a small constant factor. In particular, we provided an upper bound of 12 for the competitiveness of any online piecemeal-DFS exploration strategy. A trivial lower bound for the same problem is $3/2$ (Consider the tree with three branches of lengths $B/4$, $B/4$ and $B/2$, respectively, starting from the root: any online algorithm

may cover the tree with 3 routes, while the optimal is 2 routes). This leaves a gap between the lower and upper bounds and the interesting open question is whether the strategy $\text{PDFS}(T)$ is the best possible algorithm? Another open problem is to analyze similar strategies in other, more general, classes of graphs instead of trees.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Ambühl, C., Gasieniec, L., Pelc, A., Radzik, T., Zhang, X.: Tree exploration with logarithmic memory. *ACM Trans. Algorithms* **7**(2), 17:1–17:21 (2011)
2. Anaya, J., Chalopin, J., Czyzowicz, J., Labourel, A., Pelc, A., Vaxès, Y.: Convergecast and broadcast by power-aware mobile agents. *Algorithmica* **74**(1), 117–155 (2016)
3. Arora, S.: Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM* **45**(5), 753–782 (1998)
4. Awerbuch, B., Betke, M., Rivest, R.L., Singh, M.: Piecemeal graph exploration by a mobile robot. *Inf. Comput.* **152**(2), 155–172 (1999)
5. Awerbuch, B., Kobourov, S.G.: Polylogarithmic-overhead piecemeal graph exploration. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998*, 280–286 (1998)
6. Bampas, E., Chalopin, J., Das, S., Hackfeld, J., Karousatou, C.: Maximal exploration of trees with energy-constrained agents. *CoRR*, [arXiv:1802.06636](https://arxiv.org/abs/1802.06636) (2018)
7. Bärtschi, A., Chalopin, J., Das, S., Disser, Y., Geissmann, B., Graf, D., Labourel, A., Mihalák, M.: Collaborative delivery with energy-constrained mobile robots. In *Proceedings of Structural Information and Communication Complexity—23rd International Colloquium, SIROCCO 2016*, pp. 258–274 (2016)
8. Betke, M., Rivest, R.L., Singh, M.: Piecemeal learning of an unknown environment. *Mach. Learn.* **18**(2–3), 231–254 (1995)
9. Brass, P., Cabrera-Mora, F., Gasparri, A., Xiao, J.: Multirobot tree and graph exploration. *IEEE Trans. Robot.* **27**(4), 707–717 (2011)
10. Czyzowicz, J., Diks, K., Moussi, J., Rytter, W.: Communication problems for mobile agents exchanging energy. In *Proceedings of Structural Information and Communication Complexity—23rd International Colloquium, SIROCCO 2016*, pp. 275–288, (2016)
11. Das, S.: Graph explorations with mobile agents. In *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*. vol. 11340 of *Lecture Notes in Computer Science*, pp. 403–422. Springer (2019)
12. Das, S., Dereniowski, D., Karousatou, C.: Collaborative exploration of trees by energy-constrained mobile robots. *Theory Comput. Syst.* **62**(5), 1223–1240 (2018)
13. Das, S., Dereniowski, D., Uznański, P.: Brief announcement: energy constrained depth first search. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, Prague, Czech Republic*, pp. 165:1–165:5 (2018)
14. Dereniowski, D., Disser, Y., Kosowski, A., Pająk, D., Uznański, P.: Fast collaborative graph exploration. *Inf. Comput.* **243**, 37–49 (2015)
15. Disser, Y., Hackfeld, J., Klimm, M.: Undirected graph exploration with $\Theta(\log \log N)$ pebbles. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on discrete algorithms, SODA 2016*, 25–39 (2016)

16. Disser, Y., Mousset, F., Noever, A., Skoric, N., Steger, A.: A general lower bound for collaborative tree exploration. In *Structural information and communication complexity—24th international colloquium, SIROCCO 2017, Porquerolles, France, 2017, Revised Selected Papers*, pp. 125–139. (2017)
17. Dobrev, S., Královic, R., Markou, E.: Online graph exploration with advice. In *Proceedings of structural information and communication complexity—19th international colloquium, SIROCCO 2012*, pp. 267–278 (2012)
18. Duncan, C.A., Kobourov, S.G., Anil Kumar, V.S.: Optimal constrained graph exploration. *ACM Trans. Algorithms* **2**(3), 380–402 (2006)
19. Dynia, M., Korzeniowski, M., Schindelhauer, C.: Power-aware collective tree exploration. In *Proceedings of Architecture of Computing Systems—ARCS 2006, 19th International Conference*, pp. 341–351 (2006)
20. Dynia, M., Kutylowski, J., Friedhelm, H., Meyer auf der, Schindelhauer, C.: Smart robot teams exploring sparse trees. In *Proceedings of Mathematical Foundations of Computer Science, 31st International Symposium, MFCS 2006*, pp. 327–338 (2006)
21. Dynia, M., Łopuszanski, J., Schindelhauer, C.: Why robots need maps. In *Proceedings of Structural Information and Communication Complexity, 14th International Colloquium, SIROCCO 2007*, pp. 41–50, (2007)
22. Fraigniaud, P., Gąsieniec, L., Kowalski, D.R., Pelc, A.: Collective tree exploration. *Networks* **48**(3), 166–177 (2006)
23. Frederickson, G.N., Hecht, M.S., Kim, C.E.: Approximation algorithms for some routing problems. *SIAM J. Comput.* **7**(2), 178–193 (1978)
24. Higashikawa, Y., Katoh, N., Langerman, S., Tanigawa, S.: Online graph exploration algorithms for cycles and trees by multiple searchers. *J. Comb. Optim.* **28**(2), 480–495 (2014)
25. Megow, N., Mehlhorn, K., Schweitzer, P.: Online graph exploration: New results on old and new algorithms. *Theor. Comput. Sci.* **463**, 62–72 (2012)
26. Ortof, C., Schindelhauer, C.: A recursive approach to multi-robot exploration of trees. In *Proceedings of Structural Information and Communication Complexity—21st international colloquium, SIROCCO 2014*, pp. 343–354 (2014)
27. Panaite, P., Pelc, A.: Exploring unknown undirected graphs. *J. Algorithms* **33**(2), 281–295 (1999)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.